Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №13

Обработка символьной информации

Выполнил: Мышковец Артём Витальевич 10ПИ

Минск 2024

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Изучить способы преобразования символов, выполнив программы в правой части.  Выполнить прокрутки программ и записать условия.  Опробовать программы с различным текстом. | **#include <iostream>**  **void main()**  **{ int num = 5;**  **char symb, new\_symbol = ' ';**  **symb = num + '0';**  **std::cout<<symb<<' ';**  **if (symb >= '0' && symb <= '9')**  **num = symb - '0';**  **std::cout<<num<<' ';**  **symb = 'b';**  **if (symb >= 'a' && symb <= 'z')**  **new\_symbol = symb - 'a' + 'A';**  **std::cout<<new\_symbol<<' ';**  **}**    преобразует число 5 в символ '5' с помощью добавления ASCII-кода '0' и выводит его. Затем проверяет, является ли этот символ цифрой, преобразует его обратно в число 5 и снова выводит. После этого переменной symb присваивается символ 'b', проверяется, является ли он строчной буквой, и преобразуется в заглавную 'B', которая выводится.  Задача: Преобразовать число в символ, проверить, является ли он цифрой, затем преобразовать строчную букву в заглавную. |
| 2. Выполнить программу, приведенную в правой части.  Объяснить принцип использования *функций стандартной библиотеки*. | **#include <cstring>**  **#include <iostream>**  **using namespace std;**  **void main()**  **{**  **char str[50], s[20];**  **int i, j, i1, i2;**  **cout<<"Input string:";**  **cin>> str;**  **for(i = strlen(str); i >= 0; i--)**  **if (str[i] == '\/') break;**  **i2 = i;**  **for (j = (i2 - 1); j >= 0; j--)**  **if (str[j] == '\/') break;**  **i1 = j;**  **if (i2 == i1)**  **printf(" '\/' ");**  **else**  **{ strncpy\_s(s, &str[i1+1], i2-i1-1);**  **s[i2 - i1 - 1] = 0;**  **cout<<"katalog:"<< s<<"\n";**  **}**  **}**  **Strlen() – получаем длину строки**  **Strncpy() – копирует строки** |
| 3. Изучить способы работы с символьными массивами, выполнив программу в правой части. | #include <iostream>  using namespace std;  void main()  {  char s, t[20];  int i, j, n;  cout << "Input string: ";  gets\_s(t);  cout << "Input symbol: ";  cin >> s;  n = strlen(t);  for (i = 0; i <= n; i++)  if (t[i] == s)  for (j = i; j <= n; j++)  t[i] = t[i + 1];  for (i = 0; i < n; i++)  cout << t[i];  } |
| 4. Выполнить программу в правой части.  Закомментировать строку **cin.ignore…**  Объяснить возникшие проблемы при выполнении программы. | Поток не очищается и в буффер попадают мусорные символы  #include <iostream>  using namespace std;  void main()  {  char t[3][20];  int i, n;  cout << "Input size (<=3): ";  cin >> n;  //cin.ignore(cin.rdbuf()->in\_avail());  for (i = 0; i < n; i++)  {  cout << "Input string: ";  gets\_s(t[i]);  }  for (i = 0; i < n; i++)  puts(t[i]);  } |
| 5. В правой части записаны два варианта решения задачи. Выполнить программы и объяснить различия между ними.  Внести изменения в программу с тем, чтобы проверялось не только количество скобок, но и правильность их расстановки (первой в тексте должна быть открывающая скобка). | #include <stdio.h>  void main()  {  char s[256];  int i, count;  puts("Enter string: ");  gets\_s(s);  for (count = i = 0; s[i] != 0; i++)  {  if (i == 0 && s[i] != '(') {  printf("Wrong str!\n");  break;  }  else if (s[i] == '(') {  count++;  }  else if (s[i] == ')') {  count--;  }  }  if (!count)  puts("Ok\n");  else  puts("Not Ok\n");  } |
| 6. Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе.  При написании программне использовать стандартные функции для строк символов.  **Вариант 13**  1. Отредактировать заданное предложение, удаляя из него все слова с чётными номерами.  2. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Определить порядковый номер слова, один из символов которого находится на **k**-й позиции (если на **k**-ю позицию попадает пробел, то найти номер предыдущего слова). | #include <iostream>  using namespace std;  void deleteChars(char str[], int pos, int charCount) {  int len = strlen(str);  for (int i = 0;i != charCount;++i) {  for (int j = pos;j != len;++j) {  str[j] = str[j + 1];  }  }  }  void task1() {  char arr[300] = "Это какой-то текст для задания";  int len = strlen(arr);    int letCounter = 0;  int wordCounter = 0;  for (int i = 0;i != len;++i) {  len = strlen(arr);  if (arr[i] == ' ' || arr[i] == '\0') {  if (letCounter != 0) {  ++wordCounter;  if (wordCounter % 2 == 0) {  deleteChars(arr, i - letCounter, letCounter + 1);  ++i;  i -= letCounter;  }  }  letCounter = 0;  }  else {  ++letCounter;  }  }  cout << arr << endl;  }  void task2() {  char arr[300] = "Это какой-то текст для задания";  int k = 13;  //cin >> k;  int letCounter = 0;  int wordCounter = 0;  int len = strlen(arr);  for (int i = 0;i != len;++i) {  if (i == k) {  (\*(arr + i) != ' ' && \*(arr + i) != '\0') ? cout << wordCounter + 1 : cout << wordCounter;  }  else  {  if (\*(arr +i) == ' ' || \*(arr + i) == '\0') {  if (letCounter != 0) {  ++wordCounter;  }  letCounter = 0;  }  else {  ++letCounter;  }  }  }  }  int main() {  setlocale(LC\_ALL, "RU");  task1();  task2();  } |
|  |  |

**Дополнительные задания**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. В заданной последовательности слов найти все слова, имеющие заданное окончание. | #include <iostream>  using namespace std;  bool isThis(char str[], char okon[], int pos) {  int okonlen = strlen(okon)-1;  for (int i = pos, j = okonlen;i != pos - okonlen;--i, --j) {  if (str[i] != okon[j]) {  return false;  }  }  return true;  }  int main() {  setlocale(LC\_ALL, "RU");  char arr[300] = "Серый Белый Трава Красив Фиолетовый";  int len = strlen(arr);  char okonch[3] = "ый";  int okonLen = strlen(okonch);  int letCounter = 0;  for (int i = 0;i != len+1;++i) {  len = strlen(arr);  if (arr[i] == ' ' || arr[i] == '\0') {  if (letCounter != 0) {  if (isThis(arr, okonch, i - 1)) {  for (int j = i-letCounter;j != i;++j) {  cout << arr[j];  }  cout << endl;  }    }  letCounter = 0;  }  else {  ++letCounter;  }  }  } |
| 3. Из заданного предложения удалить те слова, которые уже встречались в предложении раньше. | #include <iostream>  #include <vector>  using namespace std;  void deleteChars(char str[], int pos, int charCount) {  int len = strlen(str);  for (int i = 0;i != charCount;++i) {  for (int j = pos;j != len;++j) {  str[j] = str[j + 1];  }  }  }  bool isThere(vector<string> unique, string word) {  for (auto item : unique) {  if (item == word) {  return true;  }  }  return false;  }  int main() {  setlocale(LC\_ALL, "RU");  char str[300] = "Пару пару пару слов будут повторяться повторяться повторяться";  vector<string> unique;  int len = strlen(str);  string word = "";  int letCounter = 0;  for (int i = 0;i != len+1;++i) {  len = strlen(str);  if (str[i] == ' ' || str[i] == '\0') {  if (letCounter != 0) {  for (int j = i - letCounter;j != i;++j) {  word += tolower(str[j]);  }  if (!(isThere(unique, word))) {  unique.push\_back(word);  }  else {  deleteChars(str, i - letCounter, letCounter + 1);  i -= ++letCounter;  }  word = "";  }  letCounter = 0;  }  else {  ++letCounter;  }  }  cout << str;  } |
| 4. Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO преобразовать в OD OD. | #include <iostream>  using namespace std;  void deleteChars(char str[], int pos, int charCount) {  int len = strlen(str);  for (int i = 0;i != charCount;++i) {  len = strlen(str);  for (int j = pos;j != len;++j) {  str[j] = str[j + 1];  }  }  }  void reverseWord(string word, char str[], int pos) {  string res = "";  for (int i = word.length()-1;i != -1;--i) {  res += word[i];  }  for (int i = pos, j = 0;i != pos + res.length();++i, ++j) {  str[i] = res[j];  }  }  int main() {  setlocale(LC\_ALL, "RU");  char arr[300] = "HOW DO YOU DO";  int len = strlen(arr);  int letCounter = 0;  int wordCounter = 0;  for (int i = 0;i != len + 1;++i) {  len = strlen(arr);  if (arr[i] == ' ' || arr[i] == '\0') {  if (letCounter != 0) {  ++wordCounter;  if (wordCounter % 2 != 0) {  deleteChars(arr, i - letCounter, letCounter + 1);  i -= ++letCounter;  }  else {  string word = "";  for (int j = i - letCounter;j != i;++j) {  word += arr[j];  }  reverseWord(word, arr, i - letCounter);  }  }  letCounter = 0;  }  else {  ++letCounter;  }  }  cout << arr << endl;  } |